



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/721,242	11/26/2003	Walter C. Dietrich JR.	YOR920030297US1	6112
48150	7590	07/03/2008	EXAMINER	
MCGINN INTELLECTUAL PROPERTY LAW GROUP, PLLC			WANG, BEN C	
8321 OLD COURTHOUSE ROAD				
SUITE 200			ART UNIT	PAPER NUMBER
VIENNA, VA 22182-3817			2192	
		MAIL DATE	DELIVERY MODE	
		07/03/2008	PAPER	

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/721,242	Applicant(s) DIETRICH ET AL.
	Examiner BEN C. WANG	Art Unit 2192

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
 - If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
 - Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED. (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(o).

Status

- 1) Responsive to communication(s) filed on 10 March 2008.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-39 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-39 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
 3) Information Disclosure Statement(s) (PTO/SB/08)
 Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
 Paper No(s)/Mail Date _____
- 5) Notice of Informal Patent Application
 6) Other: _____

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on April 10, 2008 has been entered.
2. Applicant's amendment dated March 10, 2008, responding to the Office action mailed January 10, 2008 provided in the rejection of claims 1-38, wherein claims 1, 4, 25-26 were amended, and claim 39 is new.

Claims 1-39 remain pending in the application and which have been fully considered by the examiner.

Applicant's arguments with respect to claims currently amended have been fully considered but are not persuasive. Please see the section of "Response to Arguments" for details.

Claim Rejections – 35 USC § 102(e)

The following is quotation of 35 U.S.C. 102(e) which form the basis for all obviousness rejections set forth in this office action:

Art Unit: 2192

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 23, 25, 27, 29-32, and 35-36 are rejected under 35 U.S.C. 102(e) as being anticipated by Paul A. Below (Pat. No. US 7,213,234 B1) (hereinafter 'Below')

4. **As to claim 23 (Original)**, Below discloses an apparatus to estimate at least one of a cost and an amount of necessary resources for an effort related to computer software development, computer software maintenance, and information technology services, said apparatus comprising:

- a memory to store a computer code involved in an effort related to software development; a graphic user interface to allow said computer code to be selected (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio); and
- a sampling module to allow said computer code to be sampled in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from

each of the strata; this ensures proportionate representation from each of the sub-groups in the sample)

5. **As to claim 25 (Currently Amended),** Below discloses a computer readable storage medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform a method of estimating at least one of a cost and a necessary amount of resources for an effort related to computer software development, computer software maintenance, and information technology services, said method comprising:

- reading a section of computer code;
- sampling said computer code in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample); and
- using said sampling to calculating said at least one of cost and amount of resources for a larger subset of the computer code from said computer code from said sampling (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence

interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean),

- wherein said sampling, and said calculating is executed on a computer (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio)

6. **As to claim 27 (Original),** Below discloses an apparatus to estimate a cost for an effort related to computer software development, computer software maintenance, and information technology services, said apparatus comprising:

- means for storing a computer code involved in an effort related to software development;
- means for allowing said computer code to be selected; and
- means for allowing said computer code to be sampled in accordance with a sampling technique (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio; Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random

sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample)

7. **As to claim 29 (Original)** (incorporating the rejection in claim 27), Below discloses the apparatus according further comprising:

- means for calculating cost for a larger subset of the computer code (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean) from computer code from said sampling (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample)

8. **As to claim 30 (Original)** (incorporating the rejection in claim 29), Below discloses the apparatus further comprising:

- means for calculating at least one of a risk probability and an estimation precision for said cost (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean)

9. **As to claim 31 (Original)** (incorporating the rejection in claim 27), Below discloses the method and the apparatus further comprising:

- categorizing each computer sampling into one of N categories of difficulty, N being an integer greater than 1 (e.g., Col. 3, Line 62 through Col. 4, Line 2 – In stratified sampling, the units in the sampling frame are classified into separate subgroups, or strata, on the basis of one or more important characteristics; determining which characteristics, if any, are important is somewhat of an art; the goal is to identify characteristics that have a major impact on the measurement being made; knowledge gained from previous or similar analysis can be used, as well as surveys of experts, and evidence gathered by measurement)

Art Unit: 2192

10. **As to claim 32 (Original)**, Below discloses an apparatus to estimate an amount of necessary resources for an effort related to computer software development, computer software maintenance, and information technology services, said apparatus comprising:

- means for storing a computer code involved in an effort related to software development;
- means for allowing said computer code to be selected; and
- means for allowing said computer code to be sampled in accordance with a sampling technique (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio; Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample)

11. **As to claim 35 (Original) (incorporating the rejection in claim 32)**, Below discloses the apparatus further comprising:

- means for calculating at least one of a risk probability and an estimation precision for amount of necessary resources (e.g., Col. 4, Lines 26-29 – in

general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean)

12. **As to claim 36** (Original) (incorporating the rejection in claim 32), please refer to claim 31 as set forth accordingly.

Claim Rejections – 35 USC § 103(a)

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

13. Claims 1-8, 9-22, and 37-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Ramil et al. (*Cost Estimation and Evolvability Monitoring for Software Evolution Processes, Oct., 2000, WEES 2000 Workshop on Empirical Studies of Software Maintenance*) (hereinafter 'Ramil') in view of Below.

14. **As to claim 1 (Currently Amended)**, Ramil discloses a method of estimating a cost related to at least one of computer software development, computer software maintenance, and information technology services (e.g., Sec. 1 – Introduction, Lines 1-5 – Software evolution is a continuing process that encompasses no only *ab initio* development but all activities, enhancement, adaptation or fixing, that occurs after the first operational release; however, in the present paper we consider only the post first release portion of the evolutionary process; this includes projects for enhancement and adaptation of a current software system and subsumes its maintenance, however the latter is defined; parenthetically we note that, in general and for several reasons, the term evolution is to be preferred to maintenance in the context of software for post first release activity), said method comprising:

- calculating a cost for a larger subset of the computer code from said sampling;
- wherein at least one of said reading, said sampling, and said calculating is executed on a computer (e.g., Fig. 2 – A Model-based Scheme for Change Detection; Sec. 3 – Evolvability Monitoring, 4th Par. – the general idea of such approaches is illustrated in Fig. 2; the change detection scheme, which may involve a model calibrated and believed to represent the evolution process, is run ‘in parallel’ with the real process it models; the inputs and outputs of the process are statically compared to model’s prediction; a significant change in evolvability (or other key parameter) detected by such a scheme any prompt, for example, the need for process revision, major restructuring of the evolving software or even software replacement)

Ramil discloses statistical comparison to model's prediction (e.g., Sec. 3 – Evolvability Monitoring, 4th Par.), but does not explicitly disclose:

- reading a sample of computer code in accordance with a sampling technique; and
- categorizing at least one computer sampling;

However, in an analogous art of *system and method for function point sampling for software size estimates*, Below discloses:

- reading a sample of computer code in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample); and
- categorizing at least one computer sampling (e.g., Col. 3, Lines 12-14 – Results are analyzed and quantified, and a confidence interval is determined to qualify the accuracy of the estimate; Lines 60-61 - ... allows for computation of statistical confidence levels; Col. 4, Lines 19-45 - A confidence interval for the means is an estimate interval constructed with respect to the sample mean, with a specified likelihood that the interval includes the value of the population means ... z is the value used for the

specified level of confidence ... The most frequently used confidence intervals are 90, 95 and 99 percent intervals. For situations where additional risk is an acceptable tradeoff, we use an 80 percent confidence interval; Col. 5, Line 57 – Calculate Confidence Intervals through Col. 6, Line 62 - ... Compute the confidence intervals, which can be expressed for each strata as well as for the pooled sample ... modify the confidence level and accept the additional risk that the actual result would fall outside the confidence interval ... The sample size will be driven by the precision and the level of confidence required by management)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Below into the Ramil's system to further provide the followings in Ramil system:

- reading a sample of computer code in accordance with a sampling technique;
- and
- categorizing at least one computer sampling;

The motivation is that it would further enhance the Ramil's system by taking, advancing and/or incorporating Below's system which offers significant advantages for providing a system, method, and computer program product for estimating the function point count of a software application or portfolio as once suggested by Below (e.g., Col. 2, Lines 1-13)

15. **As to claim 2** (Original) (incorporating the rejection in claim 1), Ramil discloses the method wherein said cost is for at least one of:
- porting said software to another platform;
 - maintenance of said software;
 - application portfolio management of said software; and
 - legacy transformation of said software (e.g., Sec. 1 – Introduction, Lines 1-5 – Software evolution is a continuing process that encompasses no only *ab initio* development but *all* activities, enhancement, adaptation or fixing, that occurs after the first operational release; however, in the present paper we consider only the post first release portion of the evolutionary process; this includes projects for enhancement and adaptation of a current software system and subsumes its maintenance, however the latter is defined; parenthetically we note that, in general and for several reasons, the term evolution is to be preferred to maintenance in the context of software for post first release activity)
16. **As to claim 3** (Original) (incorporating the rejection in claim 1), Below discloses the method further comprising at least one of:
- calculating a variability of said cost due to sampling error (e.g., Col. 4, Lines 26-41); and
 - calculating a probability that said cost will be lower than the cost that would have been estimated by using a sample including 100% of the code (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a

given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean)

17. **As to claim 4 (Currently Amended)** (incorporating the rejection in claim 1), Below discloses the method and the apparatus wherein said categorizing at least one computer sampling comprises:

- categorizing each computer sampling into one of N categories of difficulty, N being an integer greater than 1 (e.g., Col. 3, Lines 12-14 – Results are analyzed and quantified, and a confidence interval is determined to qualify the accuracy of the estimate; Lines 60-61 - ... allows for computation of statistical confidence levels; Col. 4, Lines 19-45 - A confidence interval for the means is an estimate interval constructed with respect to the sample mean, with a specified likelihood that the interval includes the value of the population means ... z is the value used for the specified level of confidence ... The most frequently used confidence intervals are 90, 95 and 99 percent intervals. For situations where additional risk is an acceptable tradeoff, we use an 80 percent confidence interval; Col. 5, Line 57 – Calculate Confidence Intervals through Col. 6, Line 62 - ... Compute the confidence intervals, which can be expressed for each strata as well as for the pooled sample ... modify the

confidence level and accept the additional risk that the actual result would fall outside the confidence interval ... The sample size will be driven by the precision and the level of confidence required by management)

18. **As to claim 5** (Original) (incorporating the rejection in claim 1), Ramil discloses the method further comprising:

- reading into a computer at least one of a rule by which said sampling is to be executed, and cost parameters to be used for said calculating (e.g., Sec. 2 – Cost Estimation in the Evolution Context – their parameters are to be estimated, i.e., the model calibrated, by using empirical data at various levels of aggregation)

19. **As to claim 6** (Original) (incorporating the rejection in claim 4), Below discloses the method wherein said categorizing comprises at least one of:

- a user-assisted technique in which a user enters a category for each said sampled computer code lines; and
- an automated technique in which a software tool parses each said sampled computer code line and generates a category for each parsed computer code line (e.g., Col. 3, Lines 12-14 – Results are analyzed and quantified, and a confidence interval is determined to qualify the accuracy of the estimate; Lines 60-61 - ... allows for computation of statistical confidence levels; Col. 4, Lines 19-45 - A confidence interval for the means is an estimate interval constructed with

respect to the sample mean, with a specified likelihood that the interval includes the value of the population means ... z is the value used for the specified level of confidence ... The most frequently used confidence intervals are 90, 95 and 99 percent intervals. For situations where additional risk is an acceptable tradeoff, we use an 80 percent confidence interval; Col. 5, Line 57 – Calculate Confidence Intervals (e.g., an automated technique) through Col. 6, Line 62 - ... Compute the confidence intervals, which can be expressed for each strata as well as for the pooled sample ... modify the confidence level (e.g., a user-assisted technique) and accept the additional risk that the actual result would fall outside the confidence interval ... The sample size will be driven by the precision and the level of confidence required by management)

20. **As to claims 7 (Original)** (incorporating the rejection in claim 1), Below discloses the method wherein sample is taken using at least one of:

- simple random sampling (e.g., Col. 2, Lines 9-11 – Strata are defined, and random samples are chosen for a function point count);
- cluster sampling (e.g., Col. 5, Lines 8-11 – another possibility is to use cluster sampling; cluster sampling is a type of random sampling in which the population items occur naturally in subgroups; entire subgroups are randomly sampled); and
- stratified sampling (e.g., Col. 3, Lines 62-65 – In stratified sampling, the units in the sampling frame are classified into separate subgroups, or strata, on the basis

of on or more important characteristics; determining which characteristic, if any, are important is somewhat of an art)

21. **As to claim 8 (Original)** (incorporating the rejection in claim 1), Below discloses the method wherein the sample includes at least one of:

- a line of code;
- a file or module from an application or set of applications;
- an initial part of a file or a module from an application or set of applications; and
- an application from a set of applications (e.g., Col. 3, Line 8 through Col. 4, Lines 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a compete count (census); this technique can be used to estimate size for a large portfolio as well as on large applications)

22. **As to claim 9 (Original)**, Ramil discloses a method of estimating necessary amounts of resources for an effort related to at least one of computer software development, computer software maintenance, and information technology services, said method comprising:

- calculating resources for a larger subset of the computer code from said sampling, wherein at least one of said reading, said sampling, and
- said calculating is executed on a computer (e.g., Fig. 2 – A Model-based Scheme for Change Detection; Sec. 3 – Evolvability Monitoring, 4th Par. – the general idea

of such approaches is illustrated in Fig. 2; the change detection scheme, which may involve a model calibrated and believed to represent the evolution process, is run 'in parallel' with the real process it models; the inputs and outputs of the process are statically compared to model's prediction; a significant change in evolvability (or other key parameter) detected by such a scheme any prompt, for example, the need for process revision, major restructuring of the evolving software or even software replacement)

Ramil discloses statistical comparison to model's prediction (e.g., Sec. 3 – Evolvability Monitoring, 4th Par.), but does not explicitly disclose reading a sample of computer code in accordance with a sampling technique.

However, in an analogous art of *system and method for function point sampling for software size estimates*, Below discloses reading a sample of computer code in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Below into the Ramil's system to further provide reading a sample of computer code in accordance with a sampling technique in Ramil system.

The motivation is that it would further enhance the Ramil's system by taking, advancing and/or incorporating Below's system which offers significant advantages for providing a system, method, and computer program product for estimating the function point count of a software application or portfolio as once suggested by Below (e.g., Col. 2, Lines 1-13)

23. **As to claim 10 (Original)** (incorporating the rejection in claim 9), Ramil discloses the method wherein said resources are for at least one of:

- porting said software to another platform;
- maintenance of said software;
- application portfolio management of said software; and
- legacy transformation of said software (e.g., Sec. 1 – Introduction, Lines 1-5 – Software evolution is a continuing process that encompasses no only *ab initio* development but *all* activities, enhancement, adaptation or fixing, that occurs after the first operational release; however, in the present paper we consider only the post first release portion of the evolutionary process; this includes projects for enhancement and adaptation of a current software system and subsumes its maintenance, however the latter is defined; parenthetically we note that, in general and for several reasons, the term evolution is to be preferred to maintenance in the context of software for post first release activity)

24. **As to claim 11 (Original)** (incorporating the rejection in claim 9), Below discloses the method further comprising at least one of:

- calculating one or more variabilities of amounts of resources due to sampling error (e.g., Col. 4, Lines 26-41); and
- calculating a probability that amount of resources will be less than the amounts of resources that would have been estimated by using a sample including 100% of the code (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean)

25. **As to claim 12 (Original)** (incorporating the rejection in claim 9), please refer to claim 4 as set forth accordingly.

26. **As to claim 13 (Original)** (incorporating the rejection in claim 9), Below discloses the method further comprising:

- reading into a computer at least one of a rule by which said sampling is to be executed, and resource parameters to be used for said calculating (e.g., Col. 3, Line 8 through Col. 4, Line 13; Col. 5, Lines 1-7; Col. 7, Lines 16-20 – since the stratum size is allowed to vary, this is called disproportionate stratified

random sampling; the size of the sample relative to the size of total population of applications will be used as a weighting factor in the final calculations)

27. **As to claim 14 (Original)** (incorporating the rejection in claim 9), Below discloses the method further comprising:

- creating at least one of a resource plan and a work breakdown structure based on the calculated resources (e.g., Abstract, Lines 4-6 – results are analyzed and quantified, and a confidence interval is determined to qualify the accuracy of the estimate)

28. **As to claim 15 (Original)** (incorporating the rejection in claim 11), Below discloses the method further comprising:

- creating a risk management plan based on calculated risk parameters (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean)

29. **As to claim 16 (Original)** (incorporating the rejection in claim 12), please refer to claim 6 as set forth accordingly.

30. **As to claim 17 (Original)** (incorporating the rejection in claim 9), please refer to claim 7 as set forth accordingly.

31. **As to claim 18 (Original)** (incorporating the rejection in claim 8), Below discloses the method wherein the sample includes at least one of:

- a line of code;
- a file or a module from an application or set of applications;
- an initial part of a file or a module from an application or set of applications; and
- an application from a set of applications (e.g., Col. 3, Line 8 through Col. 4, Lines 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a compete count (census); this technique can be used to estimate size for a large portfolio as well as on large applications)

32. **As to claim 19 (Original)**, Ramil discloses a business method comprising at least one of:

- estimating a cost for an effort related to at least one of computer software development and information technology (IT) services, said estimating method comprising:
- calculating said cost for a larger subset of the computer code from said computer code from said sampling (e.g., Fig. 2 – A Model-based Scheme for Change Detection; Sec. 3 – Evolvability Monitoring, 4th Par. – the general idea of such

approaches is illustrated in Fig. 2; the change detection scheme, which may involve a model calibrated and believed to represent the evolution process, is run 'in parallel' with the real process it models; the inputs and outputs of the process are statically compared to model's prediction; a significant change in evolvability (or other key parameter) detected by such a scheme any prompt, for example, the need for process revision, major restructuring of the evolving software or even software replacement)

Ramil discloses statistical comparison to model's prediction (e.g., Sec. 3 – Evolvability Monitoring, 4th Par.), but does not explicitly disclose sampling computer code in accordance with a sampling technique; calculating at least one of a risk probability and an estimation precision for said cost, wherein at least one of said reading, said sampling, and said calculating is executed on a computer; providing a result of said calculating to a party; and receiving said result of said calculating.

However, in an analogous art of *system and method for function point sampling for software size estimates*, Below discloses sampling computer code in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample); calculating at least one of a risk probability and an estimation precision for said cost (e.g., Col. 4,

Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean), wherein at least one of said reading, said sampling, and said calculating is executed on a computer (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio); providing a result of said calculating to a party; and receiving said result of said calculating (e.g., Col. 1, Lines 37-39 – end of story for process improvement and accurate, reliable operations analysis reporting).

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Below into the Ramil's system to further provide sampling computer code in accordance with a sampling technique; calculating at least one of a risk probability and an estimation precision for said cost, wherein at least one of said reading, said sampling, and said calculating is executed on a computer; providing a result of said calculating to a party; and receiving said result of said calculating in Ramil system.

The motivation is that it would further enhance the Ramil's system by taking, advancing and/or incorporating Below's system which offers significant advantages for providing a system, method, and computer program product for estimating the function

point count of a software application or portfolio as once suggested by Below (e.g., Col. 2, Lines 1-13)

33. **As to claim 20** (Original) (incorporating the rejection in claim 19), Ramil discloses the business method wherein said effort comprises at least one of:

- porting said software to another platform;
- maintenance of said software;
- application portfolio management of said software; and
- legacy transformation of said software (e.g., Sec. 1 – Introduction, Lines 1-5 – Software evolution is a continuing process that encompasses no only *ab initio* development but *all* activities, enhancement, adaptation or fixing, that occurs after the first operational release; however, in the present paper we consider only the post first release portion of the evolutionary process; this includes projects for enhancement and adaptation of a current software system and subsumes its maintenance, however the latter is defined; parenthetically we note that, in general and for several reasons, the term evolution is to be preferred to maintenance in the context of software for post first release activity)

34. **As to claim 21** (Original), Ramil discloses a business method comprising at least one of: estimating a necessary amount of resources for an effort related to at least one of computer software development and information technology (IT) services, said estimating method comprising: calculating said necessary amount of resources for a

larger subset of the computer code from said computer code from said sampling (e.g., Fig. 2 – A Model-based Scheme for Change Detection; Sec. 3 – Evolvability Monitoring, 4th Par. – the general idea of such approaches is illustrated in Fig. 2; the change detection scheme, which may involve a model calibrated and believed to represent the evolution process, is run ‘in parallel’ with the real process it models; the inputs and outputs of the process are statically compared to model’s prediction; a significant change in evolvability (or other key parameter) detected by such a scheme any prompt, for example, the need for process revision, major restructuring of the evolving software or even software replacement)

Ramil discloses statistical comparison to model’s prediction (e.g., Sec. 3 – Evolvability Monitoring, 4th Par.), but does not explicitly disclose sampling computer code in accordance with a sampling technique and calculating at least one of a risk probability and an estimation precision for said estimate of amount of resources, wherein at least one of said reading, said sampling, and said calculating is executed on a computer; providing a result of said calculating to a party; and receiving said result of said calculating.

However, in an analogous art of *system and method for function point sampling for software size estimates*, Below discloses sampling computer code in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on

large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample) and calculating at least one of a risk probability and an estimation precision for said estimate of amount of resources (e.g., Col. 4, Lines 26-29 – in general, larger samples result in smaller confidence intervals for a given probability level; the challenge is to select the smallest sample (least cost) that will result in an acceptable confidence interval; Col. 4, Lines 19-22 – a confidence interval for the means is an estimate interval constructed with respect to the sample means, with a specified likelihood that the interval includes the value of the population mean), wherein at least one of said reading, said sampling, and said calculating is executed on a computer (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio); providing a result of said calculating to a party; and receiving said result of said calculating (Col. 1, Lines 37-39 – end of story for process improvement and accurate, reliable operations analysis reporting)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Below into the Ramil's system to further provide sampling computer code in accordance with a sampling technique and calculating at least one of a risk probability and an estimation precision for said estimate of amount of resources, wherein at least one of said reading, said sampling, and said calculating is executed on a computer; providing a result of said calculating to a party; and receiving said result of said calculating in Ramil system.

The motivation is that it would further enhance the Ramil's system by taking, advancing and/or incorporating Below's system which offers significant advantages for providing a system, method, and computer program product for estimating the function point count of a software application or portfolio as once suggested by Below (e.g., Col. 2, Lines 1-13)

35. **As to claim 22** (Original) (incorporating the rejection in claim 21), please refer to claim **20** as set forth accordingly.

36. **As to claim 37** (Original) (incorporating the rejection in claim 1), Below discloses a method for deploying computing infrastructure, comprising integrating computer-readable code into a computing system, wherein the code in combination with the computing system is capable of performing the method of claim 1 (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio; Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample)

37. **As to claim 38 (Original)** (incorporating the rejection in claim 9), Below discloses a method for deploying computing infrastructure, comprising integrating computer-readable code into a computing system, wherein the code in combination with the computing system is capable of performing the method of claim 9 (e.g., Col. 3, Lines 8-10 – the preferred embodiment provides a system, method, and computer program product for estimating the function point count of a software application or portfolio; Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each of the strata; this ensures proportionate representation from each of the sub-groups in the sample)

38. **As to claim 39 (New)**, Ramil discloses a method of estimating a cost related to at least one of computer software development, computer software maintenance, and information technology services, said method comprising:

- calculating a cost for a larger subset of the computer code from said sampling, wherein at least one of said reading, said sampling, and said calculating is executed on a computer (e.g., Fig. 2 – A Model-based Scheme for Change Detection; Sec. 3 – Evolvability Monitoring, 4th Par. – the general idea of such approaches is illustrated in Fig. 2; the change detection scheme, which may involve a model calibrated and believed to represent the evolution process, is run

'in parallel' with the real process it models; the inputs and outputs of the process are statically compared to model's prediction; a significant change in evolvability (or other key parameter) detected by such a scheme any prompt, for example, the need for process revision, major restructuring of the evolving software or even software replacement)

Ramil discloses statistical comparison to model's prediction (e.g., Sec. 3 – Evolvability Monitoring, 4th Par.), but does not explicitly disclose:

- reading one or more samples of computer code in accordance with a sampling technique;

categorizing each computer sampling into one of N categories of difficulty, N being an integer greater than 1, wherein said categorizing comprises at least one of:

- a user-assisted technique in which a user enters a category for each said sampled computer code lines; and
- an automated technique in which a software tool parses each said sampled computer code line and generates a category for each parsed computer code line

However, in an analogous art of *system and method for function point sampling for software size estimates*, Below discloses:

- reading one or more samples of computer code in accordance with a sampling technique (e.g., Col. 3, Line 14 through Col. 4, Line 13 – the preferred embodiment provides a sampling system and method for estimating a portfolio size, or an application size, without the expense or time required to conduct a

complete count (census); this technique can be used to estimate size of a large portfolio as well as on large application; a random sample is taken from each the strata; this ensures proportionate representation from each of the sub-groups in the sample)

categorizing each computer sampling into one of N categories of difficulty, N being an integer greater than 1, wherein said categorizing comprises at least one of:

- a user-assisted technique in which a user enters a category for each said sampled computer code lines; and
- an automated technique in which a software tool parses each said sampled computer code line and generates a category for each parsed computer code line (e.g., Col. 3, Lines 12-14 – Results are analyzed and quantified, and a confidence interval is determined to qualify the accuracy of the estimate; Lines 60-61 - ... allows for computation of statistical confidence levels; Col. 4, Lines 19-45 - A confidence interval for the means is an estimate interval constructed with respect to the sample mean, with a specified likelihood that the interval includes the value of the population means ... z is the value used for the specified level of confidence ... The most frequently used confidence intervals are 90, 95 and 99 percent intervals. For situations where additional risk is an acceptable tradeoff, we use an 80 percent confidence interval; Col. 5, Line 57 – Calculate Confidence Intervals (e.g., an automated technique) through Col. 6, Line 62 - ... Compute the confidence intervals, which can be expressed for each strata as well as for the pooled sample ... modify the

confidence level (e.g., a user-assisted technique) and accept the additional risk that the actual result would fall outside the confidence interval ... The sample size will be driven by the precision and the level of confidence required by management)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Below into the Ramil's system to further provide the followings in Ramil system:

- reading one or more samples of computer code in accordance with a sampling technique;

categorizing each computer sampling into one of N categories of difficulty, N being an integer greater than 1, wherein said categorizing comprises at least one of:

- a user-assisted technique in which a user enters a category for each said sampled computer code lines; and
- an automated technique in which a software tool parses each said sampled computer code line and generates a category for each parsed computer code line;

The motivation is that it would further enhance the Ramil's system by taking, advancing and/or incorporating Below's system which offers significant advantages for providing a system, method, and computer program product for estimating the function point count of a software application or portfolio as once suggested by Below (e.g., Col. 2, Lines 1-13)

39. Claims 24, 26, 28, and 33-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Below in view of Ramil

40. **As to claim 24** (Original) (incorporating the rejection in claim 23), Below does not explicitly disclose the apparatus wherein said effort comprises one of: porting said computer code to another platform; maintaining said computer code; performing application portfolio management on said computer code; and performing legacy transformation on said code.

However, in an analogous art of *cost estimation and evolvability monitoring for software evolution processes*, Ramil discloses the apparatus wherein said effort comprises one of: porting said computer code to another platform; maintaining said computer code; performing application portfolio management on said computer code; and performing legacy transformation on said code (e.g., Sec. 1 – Introduction, Lines 1-5 – Software evolution is a continuing process that encompasses no only *ab initio* development but *all* activities, enhancement, adaptation or fixing, that occurs after the first operational release; however, in the present paper we consider only the post first release portion of the evolutionary process; this includes projects for enhancement and adaptation of a current software system and subsumes its maintenance, however the latter is defined; parenthetically we note that, in general and for several reasons, the term evolution is to be preferred to maintenance in the context of software for post first release activity)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Ramil into the Below's system to further provide the apparatus wherein said effort comprises one of: porting said computer code to another platform; maintaining said computer code; performing application portfolio management on said computer code; and performing legacy transformation on said code in Below system.

The motivation is that it would further enhance the Below's system by taking, advancing and/or incorporating Ramil's system which offers significant advantages which the major cost that system evolution represents in the lifetime of the system and the demands it makes on professional resources, makes it essential to be able to accurately predict, assess and control the cost of adaptation and evolution as once suggested by Ramil (e.g., Sec. of Final Remarks)

41. **As to claim 26 (Original)** (incorporating the rejection in claim 25), Below does not explicitly disclose the signal bearing medium wherein said effort comprises one of: porting said computer code to another platform; maintaining said computer code; performing application portfolio management on said computer code; and performing legacy transformation on said code.

However, in an analogous art of *cost estimation and evolvability monitoring for software evolution processes*, Ramil discloses the signal bearing medium wherein said effort comprises one of: porting said computer code to another platform; maintaining said computer code; performing application portfolio management on said computer

code; and performing legacy transformation on said code (e.g., Sec. 1 – Introduction, Lines 1-5 – Software evolution is a continuing process that encompasses no only *ab initio* development but *all* activities, enhancement, adaptation or fixing, that occurs after the first operational release; however, in the present paper we consider only the post first release portion of the evolutionary process; this includes projects for enhancement and adaptation of a current software system and subsumes its maintenance, however the latter is defined; parenthetically we note that, in general and for several reasons, the term evolution is to be preferred to maintenance in the context of software for post first release activity)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Ramil into the Below's system to further provide the signal bearing medium wherein said effort comprises one of: porting said computer code to another platform; maintaining said computer code; performing application portfolio management on said computer code; and performing legacy transformation on said code in Below system.

The motivation is that it would further enhance the Below's system by taking, advancing and/or incorporating Ramil's system which offers significant advantages which the major cost that system evolution represents in the lifetime of the system and the demands it makes on professional resources, makes it essential to be able to accurately predict, assess and control the cost of adaptation and evolution as once suggested by Ramil (e.g., Sec. of Final Remarks)

42. **As to claim 28 (Original)** (incorporating the rejection in claim 27), please refer to claim 24 as set forth accordingly.

43. **As to claim 33 (Original)** (incorporating the rejection in claim 32), please refer to claim 24 as set forth accordingly.

44. **As to claim 34 (Original)** (incorporating the rejection in claim 32), Below discloses the apparatus according further comprising: means for calculating amount of necessary resources for a larger subset of the computer code from computer code from sampling.

However, in an analogous art of *cost estimation and evolvability monitoring for software evolution processes*, Raml discloses the apparatus according further comprising: means for calculating amount of necessary resources for a larger subset of the computer code from computer code from sampling (e.g., Fig. 2 – A Model-based Scheme for Change Detection; Sec. 3 – Evolvability Monitoring, 4th Par. – the general idea of such approaches is illustrated in Fig. 2; the change detection scheme, which may involve a model calibrated and believed to represent the evolution process, is run ‘in parallel’ with the real process it models; the inputs and outputs of the process are statically compared to model’s prediction; a significant change in evolvability (or other key parameter) detected by such a scheme any prompt, for example, the need for process revision, major restructuring of the evolving software or even software replacement)

Therefore, it would have been obvious to one of ordinary skill in the art, at the time the invention was made to combine the teachings of Ramil into the Below's system to further provide the apparatus according further comprising: means for calculating amount of necessary resources for a larger subset of the computer code from computer code from sampling in Below system.

The motivation is that it would further enhance the Below's system by taking, advancing and/or incorporating Ramil's system which offers significant advantages which the major cost that system evolution represents in the lifetime of the system and the demands it makes on professional resources, makes it essential to be able to accurately predict, assess and control the cost of adaptation and evolution as once suggested by Ramil (e.g., Sec. of Final Remarks)

Response to Arguments

45. Applicant's arguments filed on March 10, 2008 have been fully considered but they are not persuasive. For examples:

In the remarks, Applicant argues that:

- a) Applicant states regarding the prior reference there is no teaching or suggestion of "... categorizing at least one computer sampling " and " ... wherein said categorizing at least one computer sampling comprises categorizing each computer sampling into N

categories of difficulty, N being an integer greater than 1" (REMARKS – page 15, 2nd paragraph)

Examiner's response:

- a) Below discloses "Results are analyzed and quantified, and a confidence interval is determined to qualify the accuracy of the estimate" (e.g., Col. 3, Lines 12-14); "... allows for computation of statistical confidence levels" (e.g., Lines 60-61); "A confidence interval for the means is an estimate interval constructed with respect to the sample mean, with a specified likelihood that the interval includes the value of the population means ... z is the value used for the specified level of confidence ... The most frequently used confidence intervals are 90, 95 and 99 percent intervals. For situations where additional risk is an acceptable tradeoff, we use an 80 percent confidence interval" (e.g., Col. 4, Lines 19-45); "... Compute the confidence intervals, which can be expressed for each strata as well as for the pooled sample ... modify the confidence level (e.g., a user-assisted technique) and accept the additional risk that the actual result would fall outside the confidence interval ... The sample size will be driven by the precision and the level of confidence required by management" (e.g., Col. 5, Line 57 – Calculate Confidence Intervals (e.g., an automated technique) through Col. 6, Line 62) (emphasis added)

Conclusion

46. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is 571-270-1240. The examiner can normally be reached on Monday - Friday, 8:00 a.m. - 5:00 p.m., EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Examiner, Art Unit 2192
June 27, 2008

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192